Data needs for securing Internet infrastructure

The GMI3S project at CAIDA

V 2.4 of 1 May 2023

Abstract

The CAIDA GMI3S (Global Measurement Infrastructure to Improve Internet Security) project has the objective of designing a new generation of infrastructure to support measurements of the Internet, a new generation of platforms and tools for data curation and utilization, and support for use of Internet measurement data by the research community. While these facilities are relevant to a wide range of measurements, the focus of GMI3S is on Internet infrastructure security. Specifically, our attention is on security vulnerabilities (and consequential harms) that arise in the packet carriage layer of the Internet. We focus on the following system components:

- The addressing architecture of the Internet, and systems to support address allocation, management, and use.
- The global routing protocol of the Internet, the Border Gateway Protocol, or BGP.
- The Domain Name System, or DNS, which maps from high-level names to IP addresses.
- The Certificate Authority system, which manages encryption keys for applications.

Our decision to focus on these elements – which we call the packet carriage service of the Internet – is motivated by three key features they share: their foundational role for all Internet use, the need for collective action to prevent harms, and the misaligned incentives to take such action.

We also consider Denial of Service attacks, DoS (or Distributed Denial of Service or DDoS), which exploit both vulnerable end nodes as well as the basic packet forwarding function of the Internet to flood a end-node or a region of the network, causing an overload that prevents proper functioning. We include DDoS in our analysis because some of the mitigations may depend on operational practices across the ecosystem, not just actions by the victim of the attack.

This document catalogs the datasets that we have identified that play a role (or *should* or *could* play a role) in improving the security posture of these underlying layers of Internet infrastructure. For each system that we survey, we summarize the known or potential vulnerabilities, and possible mitigations to these vulnerabilities. We discuss the role of data in each of these steps.

Abstract 1

1 Motivation: Securing the Foundations of Internet Infrastructure 3

1.1 The need for collective action to secure the Internet 4

1.2 Role of data in identifying, assessing, and mitigating harms 4

1.3 Mapping vulnerabilities to data 5

1.4 Barriers to action 6

1.5 The critical systems of the Internet 6

2 IP Addresses 7

2.1 Vulnerabilities and associated harms 7

2.2 Primary data 8

2.3 Derived data 8

3 BGP 9

3.1 Vulnerabilities and associated harms 10

3.2 Primary data 13

3.3 Derived data 14

3.4 Epistemological Challenges 16

3.5 Open research/questions: **Error! Bookmark not defined.**

4 Domain Name System 16

4.1 Vulnerabilities and harms 17

4.2 Primary data. 20

4.3 Derived data 22

5 Certificate Authority System 23

5.1 Vulnerabilities and harms 24

5.2 Primary data. 26

8.3 Derived data 27

9 Denial of Service attacks **Error! Bookmark not defined.**

(See other document, kc still taking pass) **Error! Bookmark not defined.**

9.1 Primary data. **Error! Bookmark not defined.**

10 General observations 37

10.1 Tactics vs. strategy 37

10.2 Need for ground truth on attacks, harms, and baselines 37

11 Bibliography 39

1 Motivation: Securing the Foundations of Internet Infrastructure

The CAIDA GMI3S (Global Measurement Infrastructure to Improve Internet Security) project has the objective of designing a new generation of infrastructure to support measurements of the Internet, a new generation of platforms and tools for data curation and utilization, and support for use of Internet measurement data by the research community. While these facilities are relevant to a wide range of measurements, the focus of GMI3S is on Internet infrastructure security. Specifically, our attention is on security vulnerabilities (and consequential harms) that arise in the packet carriage layer of the Internet. Because the Internet, as a packet carriage system, is conceptually a relatively simple system, we can identify the relevant system components of concern, and the specific vulnerabilities in them:

- The addressing architecture of the Internet, and systems to support address allocation, management, and use.
- The global routing protocol of the Internet, the Border Gateway Protocol, or BGP.
- The Domain Name System, or DNS, which maps from high-level names to IP addresses.
- The Certificate Authority system, which manages encryption keys for applications.

We also consider Denial of Service attacks, DoS (or Distributed Denial of Service or DDoS), which exploit both vulnerable end nodes as well as the basic packet forwarding function of the Internet to flood an end-node or a region of the network, causing an overload that prevents proper functioning. We include DDoS in our analysis because some of the mitigations may depend on operational practices across the ecosystem, not just actions by the victim of the attack.

Our decision to focus on these elements – which we call the packet carriage service of the Internet – is motivated by three key features they share: their foundational role for all Internet use, the need for collective action to prevent harms, and the misaligned incentives to take such action.

1.1 The need for collective action to secure the Internet

The packet carriage service of the Internet is a foundation on which every application depends. The designers of every application that operates over the Internet must consider whether and how to attempt to mitigate the harms that may arise due to poor security at the Internet layer. Poor security imposes a cost (or a risk) that every application bears.

But there is a critical difference between vulnerabilities in the end-nodes attached to the Internet and vulnerabilities in the Internet itself. Organizations that connect to the Internet can take many steps to improve their own security posture, with the help of many published best practices in user authentication, system patching, secure backup, business continuity planning, etc. Individual enterprises can assess their risk profile and invest accordingly.

In contrast, organizations that use the Internet are often not in a position to defend themselves from harms that arise from insecurity in these foundational layers of the Internet itself. Such harms may arise in parts of the Internet that are far removed from the firm being harmed, and the harmed firm may have little recourse. Mitigation of the risks to the connected firm depends on the collective action of the providers of the core Internet services. However, those actors in a position to mitigate the vulnerability often have no (or limited) incentive to take the required action. The combination of economic pressures, tensions among competing operational objectives, and problems of coordination raise formidable and persistent challenges to improved security of Internet infrastructure.

Because organizations connected to the Internet cannot defend themselves from the consequences of poor Internet security, and because achieving the necessary collective action to improve security is difficult, we focus on this challenge as a key effort in improving the overall security posture of the Internet. These security challenges are persistent, and the barriers to improvement are substantial. We believe that better data can illuminate the extent of the vulnerabilities, the potential of different proposals to mitigate those vulnerabilities, and the complexity of deploying those proposals. Thus, a guiding principle is that better visibility into the problems is the best and most urgently needed contribution to finding a way forward.

Moreover, in any scenario where collective action is required, one essential challenge is transparency regarding participation in the action. Data is required to provide such transparency. Another guiding principle is how to minimize the cost and risk of providing such data.

1.2 Role of data in identifying, assessing, and mitigating harms

Navigation of security threats occurs at five levels: prevention, tactical defense, forensic analysis, strategic mitigation, and longitudinal assessment. All of these require data, the role of which we discuss.

Prevention: The operational reality of most security threat navigation today is attempting to prevent intrusion or compromise using access control lists and/or blacklists.

Tactical mitigation: During exploitation of a vulnerability, the immediate question is how is the attacker crafting the attack from the parts provided by the ecosystem? Defenders need timely evidence of the specific attack, details of the attack and the nature of the attacker, etc. Today, tactical mitigations are typically undertaken by private actors, who often must act with uncertain authority and powers. They also usually operate without access to information that governments might obtain through a formal proceeding, but the complexity and delays of such a process are themselves barriers to tactical mitigation.

Forensic: When harms result from an attack, data is essential to assessing the harm. Observable evidence of an attack does not imply the attack was successful, or a material cause for concern. In order to assign a priority to mitigating a vulnerability, we need to establish that the resulting harms are real.

Strategic: Data informs proposed changes to systems and work flows. At first proposed changes are hypothetical. We cannot measure their behavior to see how they will mitigate vulnerabilities. Instead, we use data we have about the system, combined with our best models of how the change will affect the system, to predict the utility of the proposed mitigation. This requires data about the overall functioning of the system, including the range of benign and malicious actions. Analysts also need data to estimate the magnitude of future harms, to justify deployment of changes to the system.

Longitudinal: While defenders need tactical data in real time, analysts trying to understand trends, model the attackers and predict the magnitude of future harms need historical data. Consistent data collection over time is an essential element of strategic mitigation.

The key barrier to getting the data about the magnitude of harms, as opposed to evidence about attempted attacks, is hesitation on the part of victims to report the harm. Firms that suffer harm as a result of a cyber event typically prefer not to disclose the event. This leaves defenders struggling to make a case that the harm is important enough to prioritize among all the other issues that contend for attention. Another barrier is that the victim may not understand exactly how – or even that – the harm occurred. If a customer is redirected to a malicious web site that steals personal information, this attack may rely on a BGP or DNS hijack. The firm may be able to tell that a customer had personal information stolen, but not how. This lack of data about the methods of attacks drives ongoing disputes about the relative priority of proposed mitigations.

Worse, even knowing about malicious events does not translate into an assessment of harm. In the case of BGP, one longitudinal study revealed that certain ASes are repeatedly hijacking blocks of addresses for months if not years [3]. We cannot easily assess how much harm this is causing. Without data on actual harms, opinions differ on how important it is to mitigate this problem.

1.3 Mapping vulnerabilities to data

As part of this project, we are collecting an inventory of datasets relevant to research related to vulnerabilities in these systems. This inventory will help us understand requirements for our measurement infrastructure, the features and capacity of our platform for data curation and utilization, and opportunities for collaboration with other groups that collect relevant Internet data. This document, which we expect to evolve and grow as the project progresses, is our inventory of data.

We want to identify as many sorts of data as possible, including data that are currently collected, data that might be collected using this new generation of infrastructure, data that are collected by other groups, and data that do not now exist (or even where there is no obvious way to collect it) but which would be useful if it were possible to obtain. This planning phase also requires analysis of barriers to the collection of relevant data, both technical and non-technical. So our inventory covers data that may exist, perhaps within firms that operate parts of the Internet, but are not currently available to the research community.

To structure our search for relevant datasets, we first identify vulnerabilities in the four foundational systems: addressing, routing, naming, certificates. For each system, we summarize the known or potential vulnerabilities, and possible mitigations to these vulnerabilities. We discuss the role of data in each of these steps, and identify which data plays a role (or *should* or *could* play a role) in improving the security posture of that system.

Since a mitigation may introduce a new set of vulnerabilities, the design of actions to improve security is iterative, where a vulnerability may suggest possible mitigations, those mitigations may in turn have vulnerabilities, and so on. Improving the security of the Internet requires recognizing the dynamics of the ecosystem, in which actors adapt in response to a given adjustment.

1.4 Barriers to action

In the decentralized space of Internet operations and governance, there is often no coordinating actor with the authority to mandate a specific change in an Internet service, or even the standing to encourage a change. The Internet Engineering Task Force can create a new standard (a process which itself may fail to resolve disagreement), but the creation of a standard does not ensure its uptake. In some cases, a sufficiently powerful centralized actor can set a direction and effectively push a change into the ecosystem (for example, Certificate Transparency) but in many cases progress depends on collective decision-making and commitment. This is problematic for four reasons, aside from the fundamental challenge of misaligned incentives:

- There is often no clear agreement as to what behaviors by different actors actually constitute a malicious act (as opposed to utilizing features of the Internet as they were intended to be used, but to the disadvantage of one or another actor.)

- The Internet (and many of the malicious actions) span jurisdictions.

- There is no actor with the authority to mandate collection of relevant data.

- The Internet protocols were not designed with measurement in mind, and gathering data often depends on opportunistic methods that are at best a compromise.

1.5 The critical systems of the Internet

In the next sections of this paper, we discuss each of our four systems in turn, outlining our understanding of vulnerabilities, mitigations, and the resulting needs for data.

2 IP Addresses

2.1 Vulnerabilities and associated harms

End-point addresses are the most fundamental building block of the Internet–they identify the destination to which a packet is to go. Internet routers use the destination address to decide, at each hop across the Internet, how to forward the packet onward. Packets also contain the source address, identifying the end-point that sent the packet. This allows the recipient of the packet to reply to the sender.

Vulnerability: **Appropriation/Impersonation**: An attacker usurps addresses not allocated to that actor, and attempts to send and receive packets using those addresses. Appropriation of unauthorized addresses is often accomplished before or via a BGP hijack (See BGP section.)

Harm: The consequence is the victim will communicate with the malicious actor as if it were the intended endpoint.

Vulnerability: **Spoofing**: an attacker can "spoof" the source address in a packet, forging the source address of some other end-point rather than the actual sender. To use a spoofed IP source address to launch a DoS attack, there are two considerations. First, the attacker will want to exploit a protocol where a small query produces a large response. This gives the attack *amplification*. The protocols often used for amplification attacks include DNS, NTP, and memcached. Amplification is also possible with TCP[1] [despite that typically, the reply packet to an initial TCP SYN is another SYN, which is a small packet.

Harm: The consequence is that the receiver of the packet (the amplifier) replies to the spoofed address (the victim), sending the victim needless traffic which can overwhelm the victim. This capability is the basis for a class of (harmful) *Denial of Service* attack (See Section 9 on DOS.)

Mitigation: **Source Address Validation (SAV)**: ISPs can check the source address in the packets of their customers, and drop packets with spoofed source addresses. This procedure is described in Internet RFC 2827, BCP 38[1].

Role of data: Track compliance with BCP 38. Networks that allow packets with spoofed source addresses contribute to harms; identifying these networks publicly can spur adoption of SAV. It is also useful to be able to observe trends in SAV deployment over time, and across regions.

Incentive misalignment of mitigation: There is little benefit to an ISP if it implements BCP 38. The action may prevent some DoS attacks, but those attacks might have caused harm in a distant part of the Internet. There is a cost to implement BCP 38, mostly the operational overhead of correctly configuring and sustaining it. An ISP may not even realize that its configuration of BCP 38 has ceased to function properly, since there is no immediate feedback to the ISP if spoofed packets are originating from its network.

[1] Hell of a Handshake: Abusing TCP for Reflective Amplification DDoS Attacks Marc Kuhrer, Thomas Hupperich, Christian Rossow, Thorsten Holz. https://www.usenix.org/system/files/conference/woot14/woot14-kuhrer.pdf

2.2 Primary data

We use the term primary data to describe datasets that directly result from collection. Typically, if primary data is not collected, it is lost. Primary data is useful in itself, and as the basis for processed and derived data. Primary data can support tactical (real-time) analysis, and strategic analysis, which usually requires that the data be collected over time and archived.

Type	Source	Status	Limitations	Uses	Note
ISPs that do/do not implement BCP 38	CAIDA Spoofer (https://spoofer.caida.org)	Available (not currently funded)	Tests must run from inside the ISP, which limits coverage.	Track/verify compliance with BCP 38.	
DSAV Test	BYU https://dsav-test.byu.edu/	Operating (tool more than a data set)		This tool is designed to allow network administrators to test whether they have properly deployed DSAV---the filtering of spoofed traffic as it enters the network border	

2.3 Derived data

Derived data results from analysis and aggregation of raw data.

Type	Source	Status	Limitations	Uses	Note
The Open Resolver project	www.openresolverproject.org. Detects lack of SAV if an authoritative resolver receives a query from an address from a	Inactive	Requires DNS forwarder in an ISP that does not rewrite the source address of the query. False positives through use of sibling ASes within an ISP.	Identify Open DNS resolvers, identify ISPs without BCP 38	Maciej Korczyński operates an open resol

	different ASN than the OR-queried forwarder.				ver clon e.
Traceroute-based inference	Lone et. al: Using Loops Observed in Traceroute to Infer the Ability to Spoof (PAM 2017)	Inactive	Requires specific configuration at border routers – default route to provider, with gap in internally routed address space the ISP announced to provider. False positives through misinference of which router is a border router.	Identify networks with forwarding loops, identify ISPs without BCP 38	
IXP-based inference	Muller et. al: Challenges in Inferring Spoofed Traffic at IXPs (CoNEXT 2019)	Not available	Requires traffic data from IXPs, and accurate view of each member's customer cone.	Identify IXP members without BCP 38	

3 BGP

The Internet is made up of regions called Autonomous Systems (ASes) under independent control by their providers. Today, there are ~75K ASes that make up the Internet, most ~70K of which are *stub ASes,* i.e., that have no customers. The remainder are some form of transit service providers. The Border Gateway Protocol, or BGP, is the global routing protocol that independent networks use to exchange and process routing information that hooks these regions together to make up the global Internet. BGP messages provide those ASs the information necessary to forward packets to the final AS that hosts the destination.

How BGP Works. Each Autonomous System (AS) tells its directly connected neighbor ASes the address blocks or *prefixes* (contiguous addresses with common numeric prefix) it controls and utilizes. This step is called *originating* a BGP announcement. Each neighbor accepts and filters the announcement through its own policy, which often includes propagating that announcement to \emph{its} neighbors, so that the information propagates globally. Each AS appends its AS number to the announcement, so at any point the message includes the sequence of CASes that define the path back to the originated address block. Each BGP-speaking router uses each received AS path to re-compute its own forwarding table that specifies the ``best'' next hop to send packets to reach each destination prefix.

3.1 Vulnerabilities and associated harms

The critical security vulnerability with BGP is well-known: a rogue Autonomous System can announce a false assertion that it originates or is in the path to a block of addresses that it does not in fact have the authority to announce. BGP, as part of its design, does not include mechanisms to prevent such false assertions. Routers who accept such a false assertion will then deflect traffic intended for addresses in that block to that rogue AS, which can drop, inspect, or manipulate that traffic, or send traffic masquerading as those addresses. A malicious AS can falsify any part of a BGP announcement, including the origin prefix or AS, or the path. This attack is called a route hijack.

Vulnerability: ***BGP Origin hijack***. An attacker can falsify a route to a block of addresses, by announcing that it hosts those addresses. Such an announcement can potentially deflect traffic from its intended destination to that malicious AS.

Harm: Hijack of critical services: In particular, a malicious AS can hijack the route to a critical service element in the Internet, such as name servers (which map hostname to IP addresses), a Certificate Authority, a Regional Internet Registry, etc. A hijack causes harm by hijacking any address that makes up the eventual connection to the service in question.

Mitigation: Providers can check the origin prefix/AS announced by their customers (Route Origin Validation or ROV), thus blocking invalid *origin* hijacks from passing through that provider's infrastructure. Knowledge of ground truth can be derived from Route Origin Authorizations (ROAs), the Internet Routing Registry (IRR), or pairwise validation with the customer. Use of ROV requires that the prefix owner/operators has registered a ROA for that prefix. If so, any AS receiving a BGP announcement can choose to implement ROV and drop an invalid announcement, i.e., not just a provider and its customers.

Providers also need to check that the AS announced by their customer is legitimate for that customer.

Role of data: How many providers are performing validation of their customers' BGP announcement? How is that changing over time?

Role of data: How many ASs are dropping invalid announcements? What fraction of invalid announcements are dropped? To what extent does this limit the propagation of invalid routes across the Internet? How is this mitigation evolving over time?

Vulnerability: ***BGP Path hijack.*** Attackers use an invalid *path* announcement, which is not detected by simple Route Origin Validation.

Note:: the response to blocking simple invalid prefix attacks has spawned debate between two points of view. In one view, the fact that attackers can easily switch to this slightly more complex form of attack means that mitigating invalid prefix hijacks without also mitigating invalid path hijacks is of minimal value.[2] A contrary speculation is that invalid path hijacks will prove less practically useful for attackers because as they move through the Internet they grow longer than the valid announcement and thus are less likely to be selected by routers. To our knowledge, this debate is not resolved, and blocks forward progress.

Role of data in understanding path hijack attack surface: Are path hijacks successful, or could they be? Topological maps of the Internet, combined with hypothetical placement of victim and attacker, would enable mapping regions of potential vulnerability and perhaps harm. Additional data could further inform the analysis. Topology maps are computed and available, for example CAIDA's AS relationship data. Existing ROAs are publicly available and archived by RIPE, which allows the community to understand address space covered by ROAd. Recently research used the DROP blocklist to find an example path hijack of address space covered by a ROA in 2021.[3] Researchers can analyze archives of routing data and ROAs to determine the degree to which other attacks have been successful.

Another consideration is that most popular applications today (possible targets of a hijack) use cloud-based services that connect to the Internet at multiple points. Providers know where these points are, but this data is not generally available. To gather such data once could position probes across the Internet, and perform a DNS lookup of the application service to discover the IP address for that service in that region, known as its *catchment*. If catchment regions are small, BGP announcements to those regions will be short, and thus hard to hijack. Given the catchment map, one could analyze which attacker vantage points would be most effective. Estimation of harm would require knowledge of where users of the service are located. Imagine a U.S. bank, which has a customer base primarily in the U.S. The risk of harm to that bank from a hijack that is effective only in a distant country is probably minimal.

For any specific application, if we knew the customer base location and catchment, we could perform this analysis with some confidence. But to derive an overall assessment of harm across many typical service types on the Internet we need data on patterns of connection for a range of typical services. But the only way to understand the importance of path hijacks will be to build models of attack and mitigation based on the best available data.

Mitigation: Three proposed but not implemented/deployed mitigations to the path hijack vulnerability.

(1) BGPSec: cryptographic authentication of the entire router-level path. The IETF's Secure Interdomain Routing Working Group discussed, debated, and designed a new variant of BGP called BGPsec, finally documented in 2017 in RFC 8205. Cryptographic attestation of paths requires propagation of a new layer of cryptographic transaction at each hop, which is

[2] "ROV represents a substantial effort to get the infrastructure deployed, but without any form of AS Path protection, the level of protection offered by ROV is minimal at best. The conclusion is that ROV needs to be accompanied by some form of AS Path validation if it is to be useful." Geoff Huston, "A survey on securing inter-domain routing: Part 2", Jul 2021. https://blog.apnic.net/2021/07/09/a-survey-on-securing-inter-domain-routing-part-2/

[3] "Stop, DROP, and ROA: Effectiveness of Defenses through the lens of DROP", Leo Oliver, Gautam Akiwate, Matthew Luckie, Ben Du, kc claffy, https://www.caida.org/catalog/papers/2022_stop_drop_roa/stop_drop_roa.pdf

computationally expensive and poses a router-level (rather than AS-level or prefix-level) key distribution challenge. This scheme requires no new global authoritative database beyond the existing RPKI databases. Four drawbacks of this scheme are that every router receiving and forwarding a BGPsec announcement must perform complex cryptographic processing, every router that speaks BGPsec must must have a public key certified by a certificate authority, every AS along the path must implement BGPsec for the path to be protected, and the resulting messages are much larger than traditional BGP update messages.

(2) Providers can use (a proposed, not yet implemented or deployed) AS Provider Authorization (ASPA) database to detect invalid path announcements. This proposal requires a single global database, which itself can become the target of an attack.

(3) Our recently proposed *VIPzone (“zone of trust”)* leverages and enhances the MANRS framework to provide an incentive-compatible program that will prevent hijacks of routes of those in the program. The program requires an extensive program of BGP data collection and analysis to monitor conformance with the trustzone practices.

Role of data: To what extent is anyone using any of the proposed schemes? (How to gather this data?)

Vulnerability: ***Compromise of RIRs***: A malicious actor may attack the mechanisms (RPKI, ASPA) established to prevent hijacks. In particular, this would include attacks on the RIRs that host ROAs, or one of the Internet Resource Registries.

Role of data: To what extent are RIRs being attacked today? Are the attacks successful? Are trends observable over time? Are RIRs and similar registries exercising best practice for operating critical services?

Mitigation: RIRs should use best practices as with other critical server infrastructure..

Role of data: RIR data provide mapping from ASN to registrant.

Vulnerability: **Appropriation/Impersonation** of address space (see Section 3 IP Addresses): A rogue actor may attempt to use an AS number that is not properly registered, thus breaking the link to the registration process. Unless the transit providers check the legitimacy of the AS, the resulting vulnerability may thwart detection/analysis

.

Role of data: Are AS numbers being used that are not properly registered?

Mitigation: An RIR could revoke the AS number of an abusive AS.

Research question: Do ISPs check whether announced ASNs are properly registered?

Vulnerability: ***Misuse of revocation***: Operationalizing the practice of AS revocation would invite its use for other purposes, such as censorship. (But it already occurs today for lack of payment.)

Vulnerability: ***Malicious use of BGP communities.*** Studies have demonstrated the use of BGP communities to enable precise interception attacks[4], trigger remote blackholing, steer traffic, and manipulate routes even without prefix hijacking.[5]

[4] H. Birge-Lee, L. Wang, J. Rexford, and P. Mittal. 2019. SICO: Surgical interception attacks by manipulating BGP communities. In ACM CCS.

[5] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. 2018. BGP Communities: Even more Worms in the Routing Can. In ACM IMC.

Role of data: The core aspect of flexibility of BGP communities – ASes can dynamically assign their own local meanings and usage patterns to any given community – also makes them difficult for receivers of BGP communities to interpret and filter. Thus, network operators often choose to propagate communities that could increase the blast radius of a malicious attack. The IETF attempted to standardize aspects of BGP communities with limited success. The security research community would benefit from a dictionary of BGP community values and their interpretations, as well as automatic techniques to classify use of BGP communities in the wild.

Vulnerability: ***ROA validation software (Relying party or RP) can crash if given malformed data.***[6]

Mitigation: Perform RP resilience testing. Augment protocols with protections based on design. Establish operational practice to detect and eject delegated repositories that show malicious behavior.

3.2 Primary data

Type	Source	Status	Limitations	Uses	Note
Collected BGP route announcements	RouteViews BGP data. Source: NRSC and UOregon	Active collection. Available. Real time and historical	Given number of probe points, only limited view of total announcements across net	Detection of hijacks, deriving topology maps	
	RIPE RIS Source: RIPE NCC	Active collection. Available. Real time and historical	Given number of probe points, only limited view of total announcements across net	Detection of hijacks, deriving topology maps	
	PCH BGP data	Active collection. Available. Real time and historical	Focus on routes announced at an IXP, archives updates rather than entire routing tables. Views are “peering” rather than “full” views.	Detection of localized hijacks, understanding peering ecosystem	
	Source:Many companies: Akamai, AWS, Google, Kentik	Collected but not available.	Denser deployment of probes (perhaps 3x?) gives a more complete picture (of peering	Proprietary research	

https://doi.org/10.1145/3278532.3278557

[6] *Improving the Resiliency of RPKI Relying Party Software | RIPE Labs*, https://labs.ripe.net/author/koen-van-hove/improving-the-resiliency-of-rpki-relying-party-software/

			interconnections?).		
Assertions about valid announcements	ROAs. Source: RIRs Several: https://ftp.ripe.net/ripe/rpki/ Historical: RIPE?	Real time available, historical available	Providers may be vulnerable. Data may be erroneous.	Authoritative data to determine validity of a BGP announcement	
	IRR data. Source: many registries.	Real time available, no complete history	May be vulnerable to attack. Weak authentication (?)	Authoritative data to determine validity of a BGP announcement	
Authoritative data (“ground truth”) about valid BGP paths		Does not exist.		Validate paths in BGP assertions.	
Deployment patterns for application services	None	Not currently collected (?) See [4]	How many services would have to be mapped to get “typical” spectrum?	Allow analysis of mitigation effectiveness.	
Transit topologies for DNS name servers	DZDB (TLD zone files)	Available	Daily samples miss short attacks.	Allow analysis of resistance of DNS to route hijacks	
WHOIS database dumps	Source: RIRs	Available, not not public.	CAIDA archives quarterly.	Routing policy validation.	
AS to owner mapping	Source:RIRs	Real time available	May have issues in future about PII. How well do RIRs know their customers?	Build a model of attackers, correlate across attacks.	
Peering policies and presence at facilities	PeeringDB	Available via a public API	Data can become stale if ISP does not maintain it.	Validation of AS ownership inferences, network types, policies, and presence at facilities	

3.3 Derived data

Type	Source	Status	Limitations	Uses	Note
AS relationship	ASrank. Source: CAIDA	Active collection. Available.	Routing relationships must be inferred. Peering connections	deriving topology maps	

		Real time and historical	not alway identified due to limited underlying data.		
AS hegemony	Hegemony. Source: IIJ	Real time and historical available	Internet Health Report https://ihr.iijlab.net/		
AS interconnecftions	Hurricane electric	Available	https://bgp.he.net/		
AS to owner mapping	CAIDA AS2org	Available	Depends on underlying (WHOIS database) data that may be incomplete		
AS to owner mapping	Hurricane Electric	Available	https://bgp.he.net/		
List of ASs that drop invalid BGP announcements	https://rovista.netsecurelab.org/	?	Hard to track trends due to multiple factors influencing results.	Predict propagation of invalid routes.	
RPKI deployment state	Operators (Job Snijders?)	Public: RPKI deployment state - http://rpki.exposed/	Not maintained, 3 years old?	Related: https://rpki.exposed/ (RPKI Validator Security Issues Nov 2021 only)	
Lists of ASs that announce invalid routes	IIJ Internet Health report	Available, not currently funded.	IHR report		
Tactical blocklists	Droplist. Source: Spamhaus. Others?	Variably available.	Derived from undisclosed network monitor sources. No way to validate.	Allow ISPs to block (not forward) traffic from addresses known to be malicious.	
Announcement history	RIPE Stats: Routing	https://stats.ripe.net/	Derived from RIPE RIS BGP data		
NIST RPKI dashboard	Source: NIST	Real time and some historical		Visual representation of use of ROAd over time.	
BGP community dictionary	CAIDA	BGP Community Dictionary	Old, not maintained	Interpretation of BGP data	

		Dataset			
Valid ROA feed	RIPE	https://rpki-validator.ripe.net/ui/		Raw ROA data must be processed to confirm validity.	

3.4 Epistemological Challenges

The most often identified gaps in addressing and routing measurements are the following:

(1) The limited set of vantage points, which limit visibility of hijacks that intentionally do not propagate across the Internet, as well as visibility of many local peering links and multi-homed ASes.
(2) The incomplete set of authoritative information regarding prefix origins and intended AS paths (against which to validate BGP announcements), e.g,. IRR.
(3) Incomplete data on how popular applications are deployed across the Internet (anycast catchment).
(4) Lack of transparency into how tactical blocklists are generated. e.g., Spamhaus.
(5) Lack of knowledge of how abusive actors obtain ASNs, and whether they are complying with the RIR terms of service.
(6) Lack of awareness of features in routers that operators use to implement enhanced operational practices, and review of their default settings, to inform the behavior we see today.
(7) Lack of authoritative list of BGP hijacks especially against critical service elements: CAs, RIRs, anycast DNS platforms? Have we seen evidence of such attacks?

4 Domain Name System

The Domain Name System, or DNS, performs the essential function of translating higher-level names for endpoints (e.g., www.example.com) to the corresponding IP address. An oversimplified model of the DNS involves two stages: registration of a new name, and resolution of that name into an address. In the registration stage, the provider of a web page (or other named resource in the Internet), typically picks an available name in a top-level domain (TLD) of its choice (e.g., .com) and registers that name. A registrant looking to obtain a domain name under .com would contract with a registrar (e.g., Enom) who in turn interfaces with the registry operating .com, Verisign, to query the availability of the domain name and then claim it on behalf of the registrant. On successful purchase of a domain, the registrar is then responsible for the domain until it expires or is transferred by the registrant. In addition to contracts with the registry, registrars also have to be accredited by ICANN

The second stage occurs when a program (such as a browser) encounters a domain name (often as part of a URL) of a resource, and wants to connect to that resource, which requires resolving that name into an address. Computers attached to the Internet usually have software called a *stub* resolver which performs that task. The stub resolver normally contacts a *recursive* resolver to pursue complete resolution of the name. The recursive resolver will take each element of the domain name in turn (hence the term *recursive*) and contact the authoritative name server for that element, to find the address of the server for the next element of the domain name, and finally the address of the resource itself. Thus, given the name www.example.com, the recursive resolver will first contact the root name server to find an address for the name server (NS) for the com top-level domain, contact that name server to find the address of the name server for example.com, and then contact that name server to find the IP address of www.example.com.

Many enhancements and details make this work. For example, when a recursive resolver resolves a name (such as com) it will *cache* or remember the result, so it need not repeat the query. A name can map to another name, rather than an address, and the recursive resolver will resolve that name in turn. When the recursive resolver has found the address of the ultimate resource, it will return this value to the stub resolver as the result of the query.

Many organizations operate recursive resolvers. Most ISPs operate a recursive resolver for their customers. Large Internet firms also provide a recursive resolver as a service, including Google, Cloudflare, Quad9, and others.

4.1 Vulnerabilities and harms

The term *vulnerabilities* may not be the best word to describe some of the problems associated with the DNS; a better word might be *abusability*. The design of the DNS makes it easy for anyone to register a domain name, whether their intended use is malicious or benign. The resulting question is whether it is acceptable to use DNS as a means to thwart malicious behavior, or should it be considered a neutral component in the tension between attack and defense. Both sides are exploiting the features as a tactical element in pursuing their objectives. In this context, we review the many vulnerabilities.

Vulnerability: ***Service penetration***. An attacker may penetrate a Domain Name Server, and modify or add entries to the configured zone.

Mitigation: Operators of name servers should use well-documented best security practices related to securing host systems on the Internet.

Vulnerability/Harm: ***Identity theft***. An attacker may be able to steal the credentials of the owner of a domain name, and log in to the registrar using those credentials, effectively controlling the domain and thus any service relying on it.

Mitigation: Registrars should use robust methods to authenticate users (e.g., two-factor) and establish practices that prevent human deception (social engineering) attacks.

Vulnerability/Harm: ***Operational complexity***. DNS management and configuration complexity contributes to configuration errors, which can allow attackers to take over or manipulate those names.

Mitigation: DNS providers should provide users with clear instructions, and “correctness checkers” that inspect the configuration of their names and inform them of errors.

Vulnerability: ***User deception***. Users may be misled into using malicious tools.

Vulnerability: ***Plaintext protocols***. The basic query-response protocol of DNS is unencrypted and open to man-in-the-middle hijacks.

Mitigation: Replacing the original query/response protocol with an encrypted TCP connection will prevent modification of the communication.

Incentive misalignment of mitigation: Higher latency for DNS queries may slow the responsiveness of applications.

Vulnerability: ***Host misdirection.*** When a host first connects to the Internet, it receives the address of a recursive DNS resolver to use (usually based on DHCP), which may be a malicious or untrustworthy recursive resolver.

Mitigation: Users can manually override the default recursive resolver.

Incentive misalignment of mitigation: Most users have no idea how to do this, or which recursive resolver to pick.

Vulnerability: ***Misrouting of DNS queries***. In some parts of the world, users may be blocked from picking their own recursive resolver, and blocked from performing their own name resolution, forcing them to use an untrustworthy resolver.

Mitigation: Alternative query protocols such as DOH (DNS over HTTP) may make it harder for a restrictive regime to identify and block DNS queries.

Mitigation: An application (such as a web browser) can ignore the DNS implementation in the operating system and use its own implementation of a preferred query/response protocol and recursive resolver.

Incentive misalignment of mitigation: The recursive resolver picked by the browser may not implement the desired protections against malicious actions. Users may have no idea what protections they are receiving.

Vulnerability: **BGP hijack of DNS resolver**. The address of the intended recursive resolver can be hijacked, so the user unknowingly connects to a rogue copy of the server.

Mitigation: Use of HTTPS and proper key management can reduce this risk, but note that if the CA uses the same hijacked recursive resolver to perform domain validation, the attacker can obtain a "legitimate" certificate for the rogue web server. DNSSEC does not protect against this attack, since normally the host trusts the recursive resolver to validate DNSSEC information.

Vulnerability: **BGP hijack of name server**. The attacker may hijack an authoritative name server (see BGP discussion above), so that the user gets an answer from a malicious variant of the name service.

Mitigation: DNSSEC (not widely deployed) can provide assurance that the answer to a query is authentic.

Incentive misalignment of mitigation: The cost and complexity of deploying DNSSEC, including user confusion when it fails, has limited uptake of the protocol.

Vulnerability: **Malicious name server.** An untrustworthy authoritative name server in the chain of trust from the root name server can corrupt the returned result while preserving what appears to be a valid DNSSEC chain of trust.

Mitigation: A user can register a domain name in a TLD with a history of good behavior.

Role of Data: This attack is uncommon enough that there is limited (no?) data on DNS name servers that corrupt DNSSEC chains of trust. The only exception would be countries that deploy their own copy of the root DNS server, which can break the protections of DNSSEC for users in those countries. In such a case, the owner of the domain name has no recourse at the DNS level to protect the resolution of that name. See the discussion of the CA system below.

Vulnerability: ***Cache poisoning***. If a recursive resolver receives an incorrect response from an authoritative server, it will cache that response and use it to answer future queries until the TTL of that response expires (cache poisoning). Until that time-out occurs, users will be sent to the wrong address.

Mitigation: Use of DNSSEC can provide assurance that the answer to a query is authentic.

Vulnerability/harm: ***Operational complexity of DNSSEC***: Greatly increased complexity of configuration can lead to operator error and malicious exploits, or which cause queries to fail, which in turn causes loss of availability.

Mitigation: Provide tools for configuration and checking.

Harms: The vulnerabilities above lead to one of two undesirable outcomes. The first is that the query fails with an error message, and the associated service is unavailable. This outcome leads to frustration, loss of utility, costly complaints, etc. The second is that the user reaches the wrong IP address without knowing. If the user cannot or does not detect that this has happened (see discussion of the CA key management system below), the resulting harm can take many forms. But at the level of the DNS, the harm is that the user ends up talking to the wrong destination. Assessing the final impact of this DNS-level harm depends on many factors beyond the DNS. At a minimum, if the user is sent to the wrong destination and detects this fact, the harm is a loss of availability.

These DNS vulnerabilities arise from its initial design, where the focus was on simplicity, speed of response, and ease of implementation. Lack of attention to security has left a huge attack surface. The FIRST DNS-Abuse Special Interest Group (SIG) has undertaken an effort to organize these vulnerabilities into larger categories that can be mitigated in a systematic way.[7]

Traditional mitigations have taken three forms: hardening resolution, and adding cryptographic authentication (DNSSEC) to the query transaction, and creating (and selling) lists of malicious domains (intended to deceive/defraud the user). Disagreements on the relative importance of these approaches derive from disagreements about what the most important threats are. Resolving these disagreements without concrete data has proven intractable. However, there is consensus that these mitigations, especially DNSSEC, have created much greater implementation and configuration complexity in the system, increasing the cost to operate services such as recursive resolvers. This complexity brings new vulnerabilities, in that user and operator error create new options for attackers to corrupt the system.

One long-term reaction to these persistently unsolved vulnerabilities may be a migration away from use of the DNS for name resolution in favor of new alternatives designed with security in mind from the beginning. This outcome becomes more likely as app designers move away from web-based implementations to native application implementations. A web-based app must depend on the name resolution service provided by the browser (which can be the native implementation in the operating system or one in the browser) but a free-standing application is free to use any mechanism it wants to convert a high-level name to an IP address.

[7] https://www.first.org/global/sigs/dns/DNS-Abuse-Techniques-Matrix_v1.1.pdf

4.2 Primary data.

In contrast to BGP, the DNS has many different sorts of data. There is data about currently registered names, data about how those names are configured, data about who has registered those names, data about usage, and data about abuse. For security researchers, even knowing the registrar for a given domain, or being able to group by registrar on a set of names would be valuable, but access to this type of data requires (generally commercial/contractual) agreement across participating registries to name registrars consistently. GDPR and other privacy regulations have reduced accessibility of this data to researchers.

Type	Source	Status	Limitations	Uses	Note
Zone files (registered domain names and delegations)	Origin: ICANN and registries. OpenINTEL Dns.coffee	Historical: collected, available	Not all registries make their zone files available. One probe per day.	Overall statistics, detection of names suggesting malicious intent.	
Active DNS scan (Almost all possible DNS records)	OpenINTEL (OI).	Current, historical. Available	Coverage: 60% of namespace. Limited to CZDS and Open ccTLDs. One probe per day.	Detect changes in info from authoritative name server.	
Active DNS scan (ANY, A, AAAA, TXT, MX, CNAME')	Rapid7 fDNS	Current Historical Available	Coverage: 60% of namespace + CT Logs domains (undetermined). One probe per week.	Detect changes in info from authoritative name server.	
OpenResolver Census	Shadowserver	Accessible under agreement.	No visibility on private resolvers. Weekly.	Amplification attack surface studies.	
Short-term changes in name delegation	Active feed from registry? IFXR	Not widely implemente d (few registries)		Detect short term changes that signal attack.	
DNS traffic samples	OARC DITL. Several root servers + TLDs	Available under OARC membership agreement	Root-specific view. IPs anonymized. 24 hrs/year.	Name collisions, load on root servers.	
Passive DNS traffic data	DomainTools SIE	Accessible for non-commercial use under agreement	Coverage depends on VPs.		

Registration data (WHOIS/RDAP)	Registries, registrars.	Not currently available. Both real time and historical value.	Registrars may have incomplete information on registrants. Privacy issues limit access.	Detection of mass registrations and other suggestive actions. Punishment of malicious registrants.	
Enumeration of recursive resolvers.	APNIC Ad measurements (starting point), Root, TLD logs			Assess degree of conformance by resolvers to security practices.	
Log of queries to recursive resolvers.	Operator of resolver.	May be collected, not available. Historically useful.	Collection/sharing limited by volume of data and privacy concerns.	Track user engagement with malicious DNS names.	
Log of queries from recursive resolvers to name servers.	Source: Domain Tools	Available. Some historical data.	Data from recursive resolver gives incomplete picture of query patterns.	What names (including abusive) are actually being queried? Where?	
Evidence of malicious DNS names	Email,honeypots, etc. AmpPot, Cambridge Centre for Cybercrime	Variable.	Many different methods of propagation and sampling lead to disjoint lists.	Tactical blocking. Overall assessment of abuse level.	
Pricing information	tld-list.com	Commercial	Limited accuracy	Economic models of ecosystem	
Adoption of new protocols DOH, DNSSEC, etc.				Model improved security of DNS	
Role of DNS in various attack chains				Assess relevance of vulnerabilities.	
Evidence of successful misdirection				Assess overall consequence of these vulnerabilities.	
Estimates of actual harms.				Assess higher-level consequences.	
Ground truth	Domain owners	Not collected	No framework to collect or make available.		3

https://tcmdns.dev.dns-oarc.net/console/	CheckMyDNS (OARC)				

Notes:

1. Ground truth in this context means confirmation that the web site reached by the user is the one intended by the owner/operator of the site. The use of certificates is intended to provide this confirmation, but they are subject to attack. Verification by the domain owner (perhaps by making connections and confirming that the result is as intended) would confirm that the steps have happened properly. See section on Certificate Authority for discussion of this challenge.

4.3 Derived data

Type	Source	Status	Limitations	Uses	Note
DNS Databases	Domain Tools DNSDB	Available under agreement	Coverage.		
Zone files stats	CAIDA DZDB, dns.coffee	Available as dashboard/ API	Same coverage as available zones (see above table).		
Tactical blocklists	Spamhaus, abuse.ch, Feodo,DShield https://filterlists.com/	Variability available	Derived from undisclosed network monitor sources. No way to validate.	Allow resolvers to block (not resolve) DNS queries to malicious names.	
Registrars and registries with many abusive registrations	Source: DAAR (no names), Interisle, who else?	available	Derived from blocklist feeds, patterns of registration (limited visibility)	Bring visibility to accessories to abusive behavior.	
Names of abusive (e.g.,phishing) web sites	Large email processors	Not directly available (?)	Inferred from inspection of spam email, etc.	Generate lists used to protect users (e.g., Google Safe Browsing)	
Lists of popular web sites	Alexa, Majestic, Tranco, Rapid7, Cisco Umbrella	Variable	Varying methodology to generate lists. Considerable churn.	Useful in modeling harms, collateral damage.	
DNSSEC Stats	SWITCH https://dns-resilience.openintel.nl/statistics	Available as dashboard	Liimited to .ch and .li TLDs	DNS resilience studies	

4.4 DNS Measurement Tools

Type	Source	Status	Limitations	Uses	Note
dnsviz.net	ARCO	Public	Coverage (runs for one domain at a time)	Debugging DNSSEC config	
Zonemaster	Internetstiftelsen	Public	Coverage (runs for one domain at a time)	Debugging DNS configurations	
Hardenzie	Hardenzie.com	Public	Coverage (runs for one domain at a time)	Debugging Website configuration (DNS+HTTP+CA)	

5 Certificate Authority System

The Certificate Authority system plays a critical role in the security of Internet services: ostensibly, to provide a final check, after one endpoint has connected to another at a specific IP address, that the entity at that address is the intended one. A *certificate* is an assertion that links a domain name to a public key for that domain. The owner of the domain keeps the corresponding private key, and uses it with a challenge-response protocol that allows anyone to confirm that the domain owner has the private key. The integrity of this assertion relies on a *certificate authority* to cryptographically sign it, using *its* private key, which is in turn signed by another CA, and so on. The final signature that protects the sequence of signatures is provided by a *root* certificate authority. The public keys of the various root CAs are publicly documented, and included in software such as browsers.

If the CA system works as intended, the vulnerabilities in BGP and the DNS discussed above can at worst lead to a failure of availability. That is, while the CA system cannot ensure a connection reaches the intended destination, it can ideally detect if the connection has reached the wrong destination. Not surprisingly, this causes attackers to target the CA for malicious manipulation. As with the DNS, attacks on the CA system can result in a wide range of harms. As well, the sophisticated attacks often combine abuse or attack on a number of these systems, so whatever harm occurs cannot be cleanly associated with a specific vulnerability.

ETH Zurich has recently introduced a framework (F-PKI) to allow domain owners to define a policy to specify which CAs have authority to issue certificates for their domain name, and allow clients to choose a policy based on trust levels.[8] (This direction is promising, we need to understand the likelihood of uptake.)

5.1 Vulnerabilities and harms

Vulnerability: ***System penetration***. Attackers can penetrate a CA, take over its capability, and issue misleading certificates.[9]

Mitigation: CAs are expected to operate their systems using best practices for operational security.

Mitigation: CAs must agree to a periodic independent audit of their operational practices.

Mitigation: The CA/Browser Forum was created to review the behavior of root CAs and remove those that are deemed untrustworthy from the list of root CAs distributed in browsers and similar packages.

Vulnerability: ***Deliberate issuance of false certificates***. A CA with interests adverse to a specific service may intentionally create misleading certificates for that service, perhaps to facilitate surveillance.

Mitigation: CA/Browser forum can eject them from the set of trusted CAs.[10]

Vulnerability: ***Unexpected CAs in the list of trusted roots.*** The distributor of a computing device (e.g., a smart phone) may install an additional root certificate in the device before it is sold, allowing the controller of that certificate (the holder of the private key) to issue certificates that this device will accept.[7]

Mitigation: Vigilance by security experts can detect and publicize this action.

Vulnerability: ***Mandated interception***. Some firms are legally required to monitor employee behavior (e.g. the brokerage industry must record all conversations with clients), and as part of this may require that employees install an additional root certificate on their work computers so that the employer can intercept and decrypt the communication. Calling this a vulnerability depends on one's perspective, illustrating a fundamental tension between the goal of privacy and the goal of accountability. The question is whether/how to accommodate this interception within the design of the mechanism (which makes the mechanism explicit and easily a target of abuse) or by forcing the relevant enterprise to break the mechanism.

Vulnerability: ***Imposter names.*** When users are lured to an imposter website pretending to be a legitimate one, that website normally has a slightly different domain name. The owner of that domain controls it, and can get a valid certificate for that site. The CA system provides no protection in this case. Arguably, this is not a vulnerability of the CA system, but a reflection of an intentional design decision to limit the scope of responsibility of the CA system. The purpose of the CA system is to set up a trustworthy encrypted connection to the server identified by the domain name. It is up to some other actor to decide if

[8] F-PKI: Enabling Innovation and Trust Flexibility in the HTTPS Public-Key Infrastructure, NDSS 2022, https://arxiv.org/pdf/2108.08581.pdf.

[9] The penetration of the Diginotar CA is a well-documented case of this vulnerability. See https://en.wikipedia.org/wiki/DigiNotar

[10] https://www.zdnet.com/article/google-banishes-chinas-main-digital-certificate-authority-cnnic/

the domain name describes where the user meant to go. Evidence suggests that users cannot make this discrimination by looking at the domain name.

Vulnerability: ***Lack of user training***; Users may ignore warnings about an invalid certificate and proceed anyway, thus rendering the intended protection from the CA system ineffective.

Mitigation: Provide better tools to owners of certificates to automate management and reduce configuration errors. Provide better advice to users about the potential severity of different sorts of errors.

Vulnerability: ***Attack on certificate issuance.*** Certain attacks targeting BGP and the DNS can allow an attacker to create an invalid certificate that appears to be legitimate.

Mitigation: This vulnerability applies only to the weakest form of certificate, a Domain Validation or DV certificate. Owners of domains could choose to use stronger forms of certificates, such as the Organization Validation or the Extended Validation certificates.

Vulnerability: ***Lack of independent knowledge of certification type***. Browsers have no way to know what sort of certificate they should be receiving. If the owner has obtained an organization-validated (OV) certificate, and the attacker sends a domain-validated (DV) certificate, the browser will accept it. Browsers display information about the type of certificate to the user, but most users have no idea how to interpret that information.

Mitigation: Try to prevent the relevant attacks on the DNS and BGP. See discussion above.

Incentive misalignment of mitigation: Two of these vulnerabilities run directly into human-computer interaction challenges and incentive misalignment. The first is the problem that users ignore warnings when the browser receives an invalid certificate. Certificate management is complicated, and owners of certificates make errors that cause their certificates to be technically invalid. Users get warnings about these certificates, and are asked to decide whether to proceed. Most users do not know how to assess the risk, but choose to proceed anyway because their objective is to complete the task in question. Almost always the invalid certificate is not malicious, and there is no harm to the user. The users are thus trained to ignore these warnings, and when the user receives a warning about a real malicious certificate, they ignore the warning, thus completely eliminating the protection hypothetically provided by the CA system.

This reality illustrates a deep issue in the design of security systems. Information security is characterized as having three main goals: confidentiality, integrity and availability. The CA system is designed to detect a malformed certificate (thus in principle protecting confidentiality and integrity), by preventing the intended action from completing, thus presenting the user with a complete failure along the dimension of availability. The design does not give the user any strategy to deal with the loss of availability, except to accept the risk to confidentiality and integrity. Users observably care about availability and choose to proceed. Any mechanism that tries to prevent harm by protecting from loss of confidentiality and integrity but makes no effort to protect from loss of availability is an incomplete solution that will have many negative consequences. However, addressing the problem of availability is complicated, and difficult.

The second vulnerability is perhaps even more fundamental. Conceptually, the role of the CA system is to provide a final check that the end point making the connection has reached the intended service. In principle, it should at least turn failures at the lower layers (the DNS and BGP) into clean failures of availability. However, there is a weakness in the way Domain Validation certificates are issued that threatens this protection. To get a DV certificate, the owner of the domain must demonstrate that they have control over the domain, perhaps by installing a file on the web site. However, by hijacking the address of the web site or the address of the authoritative name server, or by penetrating the registry and changing the information about the location of the web site, an attack can deflect traffic intended for that web site to its rogue copy. By instituting this deflection and then requesting a certificate,the program doing the DV validation will perform the test against the web site controlled by the attacker. The attacker will get a certificate that looks valid in all respects.[11]

There are several lessons. One, which is well understood by attackers, is that the most vulnerable step in a security system is during initial setup, when the end points try to make an initial confirmation that they know who the other parties are. Another lesson is that the DV validation was designed to reduce the complexity of getting a certificate to encourage the use of secure connections on the web. A more complex procedure, such as (perhaps) the one used to get an OV certificate, might not be so vulnerable. However, the complexity and cost of that enhanced validation was a barrier to uptake.

The final consequence of this design is that the CA system cannot protect the users from all attacks on the DNS and BGP. To some extent, the security of each depends on the security of the other, which is a weak and unpredictable outcome. Pragmatically, the best protection is to position the name server and the service itself close to the majority of the users (to reduce the chance of effective BGP hijacks), and to put strong operational practices in place to reduce the probability of a social engineering attack on the staff of the organization owning the domain to prevent theft of their registry/registrar login credentials. A domain owner who has their login credentials stolen is vulnerable to a wide range of malicious consequences.

5.2 Primary data.

Type	Source	Status	Limitations	Uses	Note
Certificates in active use	Scan of Internet to find web sites. source: Censys	Query interface available. History?			
Certificates logged in certificate transparency logs	Is someone scraping these logs?	History?			
Lists of trustworthy and untrustworthy root CAs	CA/Browser forum. Browser-specific files	Real time available. History?			
Birth/death of CAs.					
Ground truth on					

[11] To some extent, Certificate Transparency can help mitigate this problem, but the extent of the actual protection is not clear.

successful and unsuccessful attacks.					
Certificate data collection:	Parse CT logs, get domain names, collect CAA and A records		See whether CAA record confirms CA issuance.		

Notes:

5.3 Derived data

Type	Source	Status	Limitations	Uses	Note
Which CAs and root CAs are used to obtain certificates			Static analysis.		
Which CAs and root CAs show up in queries.			Dynamic analysis.	What harms occur from excluding untrustworthy CAs?	
Data about dynamics of cert creation.					
Data about CA operational failures	See [5]				

6 Denial of Service attacks

Our discussion of Denial of service (DoS) attacks is different in character from the previous sections, which looked at specific systems that constitute the “transport plumbing” of the Internet. Here we discuss a class of attacks that leverage fundamental aspects of these systems, most notably that routers will make their best efforts to forward all traffic to the destination IP address in the packet, regardless of the purpose of the traffic.

The term DoS covers a wide range of attacks, with different structure and strategy. Given that the focus of the GMI3S project is on security at the Internet layer, we need some criteria to identify DoS attacks that are within the scope of this study. We limit our focus to DoS attacks that either:

- Exploit a feature of an Internet level service as a part of crafting the attack.
- Attack an Internet service using features or vulnerabilities of that service.
- Have an impact on the Internet layer itself.
- Can be detected and/or mitigated at the Internet layer.

One way to organize our discussion of attacks is to distinguish between simple flooding attacks and attacks that degrade a service by exploiting a specific feature of that service–a feature that in this context might be called a vulnerability. An example of the latter attack is the well-known SYN-flood attack, where an attacker sends TCP SYN packets, each of which induces the allocation of a block of memory: the Transmission Control Block (TCB) associated with an active TCP connection. A flood of SYNs can exhaust the supply of TCBs, preventing the victim from accepting a legitimate request to open a TCP connection.

Many attacks that exploit a feature/vulnerability of a service, e.g., SYN-flood, can be characterized as *state exhaustion* attacks. Any protocol or mechanism where an incoming message causes an allocation of a resource to create a stateful record can be vulnerable to a state exhaustion attack. Every level of the protocol stack has design features that create a vulnerability to a state exhaustion attack, but many such attacks are outside the scope of this study, based on the four criteria above. In particular, state exhaustion attacks often need a much lower rate of attack packets than a brute-force flood, and may have no observable impact on traffic. As an example, the so-called “slow loris” attack tries to bog down a web server by sending packets, each of which contains a few more bytes of a GET request, and sending them as slowly as possible, but just fast enough that the receiver does not timeout and reclaim the resources holding state information for the request. In this case, the state exhaustion attack succeeds by sending slowly, which minimizes impact and visibility of the attack at the Internet level.

Another form of attack exploiting a feature of a service tries to exhaust the processing resource of the service by sending a query that requires significant processing. An example is the “slow drip” attack against a DNS authoritative name server, in which the attacker sends many requests to resolve a different invalid subdomain of a second level domain name (SLD). Recursive resolvers will not have a cached reply to such requests, and will forward them to the appropriate authoritative name server, which may not have the resources to deal with this flood of requests.

Another way to organize our discussion is by structure or method of attack, e.g., *reflection* and *amplification*. In *reflection*, an attacker sends a packet to an intermediate service with a falsified source address, which causes that service to send a reply to that address, which is actually the final victim. In *amplification,* the attacker crafts a request to that intermediate service that triggers a reply that is larger than the request, so that the rate at which bytes arrive at the final victim is larger than the rate at which the attacker must send the stream of requests. Reflection attacks exploit the fact that ISPs only inconsistently implement Source Address Validation, so attackers can send packets with a forged source address. Amplification attacks exploit specific features of network services, which may (or may not) be essential to their normal operation.

Today, the two most exploited network services used in amplification attacks are the DNS and the Network Time Protocol (NTP). The NTP request enabling the most amplification is the “get monlist” request, which returns the identity of the last N time requests, which might be very large. This request was not a part of the normal operation of NTP, but rather more of a debugging tool. The mitigation of this vulnerability is to turn off the monlist feature.

A common exploit using DNS queries is to send queries that trigger larger replies. Attackers scan to find names that trigger large replies, and query for these to amplify an attack, or to exhaust the resources of the server.

A denial of service attack can also be *distributed* (DDoS) – originating from many locations. The most common DDoS attack method exploits a *botnet:* a set of devices attached to the Internet that a malicious actor controls, and may sell access to for use in DDoS attacks (and other attacks beyond the scope of this project). Packets used to create and manage the botnet (*command and control* traffic) may be observable in the network.

False or "spoofed" source addresses play another role beyond enabling a reflection attack. In attacks that do not require a valid response from the victim, including brute-force flooding attacks, the attacker can put a random source address into the attack packet, thus hiding its identity. Hiding the identity is of less concern to an attacker that is using a botnet, since it is hard to associate the devices with the attackers. But if an attacker is using machines that defenders can associate with the attacker, falsifying the source address provides a level of protection. Attackers may randomly assign a different spoofed address to each packet to further disguise their activities. Attacks that exploit this technique are called Randomly Spoofed DoS (RSDoS) attacks. Unlike reflection attacks, the malicious traffic is sent *directly* from the attacking infrastructure towards the victim.

6.1 Vulnerabilities and harms

The discussion of vulnerabilities in this section does not follow the pattern of the previous sections, because the vulnerabilities we identify are not part of a specific service, but are vulnerabilities in parts of the Infrastructure that facilitate the effective use of DoS attacks.

Vulnerability: A sender can falsify the source address in a packet, making it seem to come from another source.

Harm: Attackers can disguise attacks and thus make it harder to detect and mitigate them.

Mitigation: Encourage implementation of Source Address Validation (SAV) as a best practice, and sustain activities to measure compliance.

Mitigation: Eliminate all "single packet" interactions on services that attacks can exploit for amplification attacks, and replace them with interactions that require a handshake (as in the TCP SYN).

Mitigation: Redesign anycast to implement Reverse Path Forwarding (RPF) in a similar way multicast does. Symmetrical routing helps in eliminating spoofed sources. Special address space or AS identification can be required to host symmetrical services.

This step prevents the service from enabling reflection attacks, because the sender must use its actual source address to complete the handshake.

Incentive misalignment of mitigation: This design approach adds at least one round trip to the request/response, which reduces performance.

Vulnerability: The above mitigation may require creating state on the service, which opens the service up to a state exhaustion attack directed at the service. Stateless handshakes (such as SYN cookies) can be used to further mitigate this vulnerability.

Vulnerability: Network services implement request/response pairs where the response is larger than the request. This pattern is often necessary to operate the service.

Mitigation: Redesign service to eliminate request/response patterns. (May not be feasible.)

Mitigation: Limit accessibility of service, e.g., to local networks.

Mitigation: Increase cost to the querying party, e.g., require multiple sub-requests. Require that queries that will trigger large responses use TCP.

Mitigation: Require initial handshake to prevent use as part of amplification attack. Use stateless handshake such as a SYN cookie to avoid creating a state exhaustion vulnerability.

Incentive misalignment of mitigation: As above, these mitigations add latency to the request/response, which reduces performance.

For example, the DNS traditionally uses single round-trip UDP request-response interactions, which optimizes performance by removing round trips from the query. Privacy concerns have inspired new proposals to replace the UDP-based protocol between stub and recursive resolvers with either TCP-based protocols such as DoT (DNS over TLS) or DoH (DNS over HTTPS) or QUIC-based protocols [RFC9250]. QUIC, based on UDP, introduces an anti-amplification factor of 3x. Deprecating the use of UDP for DNS, or requiring a multi-round UDP handshake such as DNS over QUIC (DoQ, RFC 9250) would change amplification factors, but the risk from spoofed DDoS attacks remains..

Today, most recursive resolvers also use the UDP protocol to query an authoritative name server. But if the query from the stub to the recursive resolver required a handshake and thus could not be spoofed, then use of UDP to the authoritative name server does not create a new reflection vulnerability.

We do not discuss Network Time Protocol (NTP) in this document, but its protocols are UDP-based single packet request/responses. Other protocols allow UDP-based amplification attacks.[12] The primary mitigation here is to eliminate responses that provide amplification.

6.1 Primary data.

We use the term primary data to describe datasets that directly result from collection. Typically, if primary data is not collected, it is lost. Primary data is useful in itself, and as the basis for processed and derived data. Primary data can support tactical (real-time) analysis, and strategic analysis, which usually requires that the data be collected over time and archived.

In this table we list both data that is relevant to DoS attacks, and data related to botnets, since botnets serve as a vector to launch DoS attacks and may be observable in the network with appropriate monitoring vantage points.

[12] Amplification Hell: Revisiting Network Protocols for DDoS Abuse, https://www.christian-rossow.de/publications/amplification-ndss2014.pdf

Type	Source	Status	Limitations	Uses	Note
Network telescope Raw traffic traces in pcap format (passive monitor of unused address space.)	UCSD Network Telescope	Accessible under CAIDA agreement via Swift Storage https://www.caida.org/about/legal/aua/ https://www.caida.org/about/legal/aua/telescope_aua/ CAIDA UCSD maintains a two-month sliding window of the most recent data. Older trace files can be accessed by request	Huge size, Each pcap file contains 1 hour of data and is typically over 100 GB large Unanonomized, not truncated, meaning privacy implications Limited to Randomly Spoofed attacks. Low visibility on multi-vector attacks	Detection of scanning. Detecting ongoing DDoS attacks through "backscatter".	
Network telescope Live Feed of the Traffic (nDAG)	UCSD Network Telescope Live feed of the traffic	Accessible under CAIDA agreement https://www.caida.org/about/legal/aua/ https://www.caida.org/about/legal/aua/telescope_aua/ two streams of packets: raw packets and tagged packets. Raw packets are the original data and are formatted exactly like pcap data, while tagged packets	libtrace analysis programs that are used with the nDAG feed should be pre-configured to run eight processing threads of their own No historical data	Detection of attacks and scans in real time	

		contain more information through processing.			
Network telescope	Merit’s ORION Network Telescope (wiki)	Accessible under Merit’s data usage agreements. Availability of data in Google BigQuery (see wiki link for format) as well as PCAP data stored at Merit (going back to 2005)	Unanonomized, not truncated, meaning privacy implications BigQuery cost	Detection of scanning. Detection of RSDoS attacks through “backscatter”. Data can be easily analyzed using standard SQL queries	
UDP protocol-aware honeypot (AmpPot)	CISPA (Yokohama National University)	Live Data feed and historic DDoS data (all amp attacks since 2015). PCAP of recent amp requests	Focusing on amplification attacks only.	Detects ongoing attacks using protocol aware honeypot.	
UDP protocol-aware honeypot (HopScotch)	Cambridge Cybercrime Centre	Accessible under agreement. Live Feed and historical Data available	Data sharing process: https://www.cambridgecybercrime.uk/process.html	Detects ongoing attacks using stateless honeypot UDP reflectors, on victim hosts as well as the authoritative DNS infrastructure	
Self-reported booter usage data	Cambridge Cybercrime Centre	Accessible under agreement	Data sharing process: https://www.cambridgecybercrime.uk/process.html	Many booters report usage data. This is collected on a weekly basis	
Vulnerable device honeypot				Capture malware samples, botnet behavior.	

Reflected DDOS victims (UDP packets in pcap format)	Cambridge Cybercrime Center	Accessible under agreement:	Data sharing process: https://www.cambridgecybercrime.uk/process.html		
Active scanning	Censys			Identification of vulnerable machines to estimate potential scale of botnet. Detection of services with potential for amplification (e.g., open DNS resolvers).	
Passive packet capture	<What to list> Canadian Institute for Cybersecurity attack traces.		What can be seen depends on where the monitor is: e.g., transit link vs. enterprise access.	Real time detection of botnet operation and DoS attacks. Training/evaluating attack detection tools. Tracking activity over time.	
Passive DNS query capture		Farsight/Domain Tools	If collected between recursive and authoritative server, caching distorts observed rates. If collected between stub and recursive resolver, high volumes and privacy issues.	Part of botnet C2 detection. Detection of DoS attacks on DNS.	

Active DNS resolution	OpenINTEL			Finding options for reflection and amplification attacks (e.g., amplification potential of domain names)	
Data from DoS mitigation services	Mostly commercial		May be proprietary	Track levels of activity and evolution of methods.	
Blackholing events at IXP scale	DE-CIX Brandenberg University https://dl.acm.org/doi/10.1145/3544216.3544268#sec-supp	Paper published https://dl.acm.org/doi/10.1145/3544216.3544268 IXP Scrubber: Learning from Blackholing Traffic for ML-Driven DDoS Detection at Scale	Proprietary	Detecting ongoing attacks using blackholing inference events at IXP level	
Passive ccTLD DNS traffic (from TLD authoritative)	SIDN (.nl)	Historical data available internally	SIDN employees must be involved	Detect DDoS attacks on .nl domains	
DDoS fingerprints (JSON summaries)	NBIP.nlhttps://nbip.nl/ DDoS Clearing House · GitHub	Upon request (SIDN has contacts). 269 unique attacks from Mar-Jun 2022	Private data from members of scrubbing service	Pilot running in the Netherlands (SIDN collaboratively inclined)	
Malware sandbox (Linux + Windows)	CISPA	Closed (no APIs available; rather manual process)		Executes malware and observes network communication	
Malware	Spoki (HAW, FUB)	Upon request		Downloaders, and binary files and shell scripts to which downloaders of two-phase TCP scanners refer to.	
Curated B-Root Events	ANT	https://ant.isi.edu/datasets/all.ht		Replay to test defenses; examine to	

(including DDoS)		ml		construct defenses	
Synthetic DDoS Events	ANT	https://ant.isi.edu/datasets/all.html		Controlled-strength attacks idea to set detection sensitivity.	

6.2 Derived data

Type	Source	Status	Limitations	Uses	Note
Network telescope Randomly and Uniformly Spoofed Denial-of-Service (RSDoS) Attack Metadata	CAIDA UCSD UCSD Network Telescope (see above)	Accessible under CAIDA Agreements https://www.caida.org/about/legal/aua/ https://www.caida.org/about/legal/aua/telescope_aua/ Ongoing hourly data Available in avro format since 2020-07-14, in csv format since 2008-10-01	Randomly Spoofed attacks only	Longitudinal studies of spoofed DOS attacks	
Network telescope aggregated flow data (Flow Tuple, avro	CAIDA UCSD UCSD Network Telescope (see	Accessible under CAIDA agreements ,	Contains only certain fields: IP, dest IP, source port, dest port and	Detecting ongoing RSDoS attacks, worms	See Acceptable Use

format)	above)	https://www.caida.org/about/legal/aua/ https://www.caida.org/about/legal/aua/telescope_aua/ Ongoing hourly data, available since 2003-11-06	transport protocol + additional header fields (e.g., TCP flags)		Agreements:
Network Telescope Time Series Data	UCSD Network Telescope (see above)	Accessible under CAIDA agreement, https://www.caida.org/about/legal/aua/ https://www.caida.org/about/legal/aua/telescope_aua/ Ongoing data since 2020-04-17 accessible via Grafana Dashboards	Limited # of variables packets per second; bits per second; unique source IPs per minute; unique source ASNs per minute; unique destination IPs per minute	Detecting changes over time	
Amplification attack map	CISPA	Accessible after registration		Shows live feed of amp targets on a map	
Amplification DDoS fingerprints	CISPA	API available		Maps amplification attacks to scanner IPs	
DDoS fingerprints	NBIP.nl DDoS Clearing House · GitHub	Upon request (SIDN has contacts). 269 unique attacks from Mar-Jun	Private data from members of scrubbing service	Pilot running in the Netherlands	

		2022			
Aggregated packet trace data			May raise privacy issues.	Detect attacks by correlation across monitors.	

7 General observations

7.1 Tactics vs. strategy

Data is used in at least two ways in the ongoing practice of security. One is to thwart attacks as they are happening; the other is to evolve the system to make attacks harder to undertake. Tactical defenders need data in real time, often almost instantly. The blocking of domain names associated with phishing sites illustrates this battlespace. Evolving the system will necessarily happen more slowly, based on collective understanding of how attacks exploit the system and what the scope of action is for the attackers.

For example, the CA system cannot protect users from a domain name crafted to confuse and deceive the users by its similarity to a familiar and trustworthy name. Further, there is evidence that users cannot reliably distinguish between valid and imposter names [6]. In this context, Google has developed their “safe browning” capability, which uses their massive intake of email to detect spam email containing malicious URLs, and warns users that attempt to go to those sites. The safe browning capability still needs to identify malicious URLs as quickly as possible, but seems to make better use of the knowledge than previous mitigations. Google reports that they are delivering 3-4 million warnings to users per week,[13] down from peaks above 60 M per week at times during the previous decade. Does this mean that the “phishing problem” has been solved, or are the attackers just evolving their methods? Only ongoing and creative data collection can answer that question.

7.2 Need for ground truth on attacks, harms, and baselines

Several of the lists above included data on authoritative information, or “ground truth”. One problem that limits both analysis and mitigation of vulnerabilities is lack of reliable data about what is happening. The logic of attack and defense has two stages: what are attackers doing now, and what might they do if their current practices are degraded? If their attacks are currently successful, then we should assume that attackers will adapt to changing mitigations put in place to thwart them. For this reason, while it is sometimes useful to track the level of attack, a more fundamental result would be to

[13] https://transparencyreport.google.com/safe-browsing/overview

measure the extent of harm. However, doing so is problematic. First, those that are harmed may not wish to report the event, or may not be aware of how to report the event. (The FTC has a website where victims of fraud and other sorts of harm can report them, but what fraction go unreported?) Second, it is not always clear what sort of attack caused the harm.

Ideally, the operators of the Internet would put in the effort to clean up their part of the system, but doing so is costly, and the benefit of the cleanup will not always accrue to them. So without a strong argument that the effort is justified, it is hard to make progress.

It is highly likely that the success rate of these attacks is low. If the cost of attack is low, then even a low success rate justifies the continued attacks. If the cost of mitigation exceeds the magnitude of the harms, the best approach may be to leave the Internet as it is, and manage the harms through a risk-sharing approach such as insurance. We lack the basic data to assess these options.

The more hypothetical aspect to the calculus of attack and defense is to consider what might happen tomorrow. Could attackers exploit these vulnerabilities in new ways that greatly increases harm? The emergence of ransomware is an example of a new and more costly kind of harm that exploits the same vulnerabilities as previous malicious undertakings. Since it takes time to put defenses in place, and the vulnerabilities outlined above are well known, should the Internet operators have a duty to mitigate them, as part of overall Internet hygiene?

While operators are willing to some extent to undertake improved practices that improve security, as illustrated by the growth of the Mutually Agreed Norms for Routing Security (MANRS) initiative, compliance with such practices remains understudied. If compliance is costly, the motivation may be low. In a context like this, regulation may be justified, because it puts the burden on all operators, so that operators that comply can assume that their competitors are also bearing the cost. However, making the case for regulation calls for data about actual harms, or a compelling argument about the potential for increased harms in the future.

We summarize what we know, and what we would like to know, about the various cases above.

BGP hijacks are not always caught by the BGP route collectors, if the hijacks have a small scope. Hijacks with small scope may cause little harm, but there is no evidence to support or refute that guess. One common use today for hijacks with low scope is to launch a massive spam email campaign. Perhaps the best way to mitigate this harm is within the email system, not at the routing system. What data could help answer that question?

We don’t know how often users are harmed when they choose to ignore a warning about an invalid certificate. Is the problem serious enough to make its mitigation a high priority? The most common reason a certificate is invalid is that it has expired. What is the actual risk calculus in using a certificate that has expired? Why would an attacker send such a certificate? When a potential victim is lured to a malicious web site, does that web site normally return a valid certificate? When the certificate is invalid as part of an actual attack, what does it look like?

In some cases it is possible to count the number of attacks. For example, by counting the domain names of imposter websites that show up in phishing emails, defenders can estimate the number of active phishing sites in use at any time, and thus the number of abusive registrations of domain names. In 2020, Google estimated that there were perhaps 2M such sites. However, we don’t know how often victims were successfully lured to these sites, or what the resulting harm was. The FTC collects reports of fraud, and categorizes them by type of attack (including phishing attacks).[14]

Baseline operations. Ground truth about normal, benign operation is equally as important to the practice of defense as data about malicious activity. Many detection systems today suffer from a high rate of false positives, because the system cannot distinguish between a perhaps unusual but benign action and a malicious one. Dealing with these false positives is time-consuming, confusing to users, and a barrier to making a case for investing in improved mitigation.

By collecting data and modeling what innocent users do, we can sharpen our understanding of the distinction between benign and malice, and help avoid the false positives.

An important topic of study is the hopefully minor operational errors that trigger warnings. Can we gather data that helps us distinguish between these errors and actual malice, or can malicious actors disguise their behavior as fitting into the profile of operational errors. As an example, is a certificate that has passed its expiration date likely to be a harmless error, or can this error be exploited to mask a real attack?

What appear to be BGP hijacks may be benign operations, or may be operational errors that need to be corrected (perhaps rapidly) but are not a signal of malice.

11 Bibliography

[1] D. Senie and P. Ferguson, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,” Internet Engineering Task Force, Request for Comments RFC 2827, May 2000. doi: 10.17487/RFC 2827.

[2] E. Rye, R. Beverly, and K. C. Claffy, “Follow the scent: defeating IPv6 prefix rotation privacy,” in *Proceedings of the 21st ACM Internet Measurement Conference*, New York, NY, USA, Nov. 2021, pp. 739–752. doi: 10.1145/3487552.3487829.

[3] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark, “Profiling BGP Serial Hijackers: Capturing Persistent Misbehavior in the Global Routing Table,” in *Proceedings of the Internet Measurement Conference*, Amsterdam Netherlands, Oct. 2019, pp. 420–434. doi: 10.1145/3355369.3355581.

[4] N. P. Hoang, A. A. Niaki, M. Polychronakis, and P. Gill, “The web is still small after more than a decade,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 50, no. 2, pp. 24–31, May 2020, doi: 10.1145/3402413.3402417.

[5] N. Serrano, H. Hadan, and L. J. Camp, “A Complete Study of P.K.I. (PKI’s Known Incidents),” *SSRN Electron. J.*, 2019, doi: 10.2139/ssrn.3425554.

[6] C. Thompson, M. Shelton, E. Stark, M. Walker, E. Schechter, and A. P. Felt, “The web’s identity crisis: understanding the effectiveness of website identity indicators,” 2019, pp. 1715–1732.

[7] N. Vallina-Rodriguez, J. Amann, C. Kreibich, N. Weaver, and V. Paxson, “A Tangled Mass: The Android Root Certificate Stores,” in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, Sydney Australia, Dec. 2014, pp. 141–148. doi: 10.1145/2674005.2675015.

[14] Explore FTC data. https://www.ftc.gov/news-events/data-visualizations/explore-data